



GitHub

Transcripción

[00:00] Hola todos y todas. Bienvenidos y bienvenidas una vez más a este curso de Git. En el último vídeo conseguimos sincronizar los contenidos de los archivos de Ana y Bruno. Pero surgió la siguiente cuestión.

[00:12] ¿Necesito tener un servidor en mi red o una carpeta compartida? O incluso si voy a hacer un proyecto personal, Ana y yo somos amigos y no tenemos plata para hospedar nuestros archivos, ¿hay alguna alternativa para crear servidores gratuitamente y además que cada uno de nosotros pueda compartir los archivos a través de internet, cada uno desde nuestras casas?

[00:33] Si sabes a lo que quiero llegar probablemente ya te diste cuenta que estoy hablando principalmente de GitHub. Claro que existen otros servicios pero nosotros vamos a enfocarnos en GitHub. Y GitHub nos permite justamente crear repositorios git en internet. Entonces vamos ir a GitHub.com. Esta es la pantalla principal que nos presentan. Nosotros podemos crear una nueva cuenta o loguearnos.

[01:07] En este caso yo ya tengo una cuenta creada acá en GitHub. Entonces voy a loguearme. Tengo activado el doble factor de autenticación, así que enseguida vuelvo. Después de colocar el factor de autenticación, esta es la pantalla que se nos presenta. En este caso como yo ya tengo una cuenta creada, acá se presentan varias informaciones.

[01:35] Pero ustedes si acaban de crear una cuenta nueva, posiblemente tengan menos cosas acá. Lo que sí, independientemente de en cuál pantalla ustedes

estén, por ejemplo si yo me voy al GitHub, a alguna otra pantalla, siempre van a tener este botón + acá arriba a la derecha y con la opción para crear un nuevo repositorio.

[01:59] Entonces vamos a new repository y acá tenemos quien es el dueño de ese repositorio, en este caso yo solamente tengo mi usuario, pero por ejemplo si perteneciera a una red, a un grupo de personas, podría tener otras personas acá. Acá nosotros podemos poner cualquier nombre que queramos, en este caso yo lo voy a llamar de Alura-git. Vamos a poner una descripción, en este caso voy a poner: Lista de cursos para controlar con Git.

[02:42] Acá tenemos la opción de que sea público o privado. Anteriormente GitHub no permitía crear servidores privados para cuentas gratuitas, tenían que pagar un plan mensual, pero desde 2020 ya se consigue crear repositorios privados con colaboradores ilimitados. Así que pueden colocar privado o público, no hay problema.

[03:08] Una vez que dan a Create repository, va a demorar acá un poco y esta es la primera pantalla que nos aparece cuando creamos el repositorio. Entonces por ejemplo acá nos está diciendo que en nuestra terminal podemos hacer un git init, que nosotros ya lo hemos hecho, un git add de los archivos, un commit con la frase y acá ir a una nueva rama, agregar el repositorio origin y hacer un push.

[03:45] ¿Qué es lo que ocurre? Nosotros ya tenemos archivos en nuestro repositorio localmente, entonces no es esto lo que nosotros queremos hacer. Lo que nosotros queremos hacer es esta parte de acá, hacer un push a un repositorio ya existente. Nos está diciendo: "Es posible que ustedes tengan activada esta opción de SSH".

[04:09] Si es así, este primer comando va a hacer algo parecido con esto, que nos está diciendo que los loguemos dentro, utilicemos el comando para agregar el repositorio Git, con un usuario Git y una sintaxis acá rara que no

conocemos por ahora, entonces si utilizáramos esta opción tendríamos que agregar una aquí, con SSH y una cuestión de seguridad que en nuestro caso no es necesario. Nosotros podemos directamente colocar acá en HTTPS acá arriba.

[04:50] Y acá se simplifica bastante más el comando, en el sentido de que cada vez que nosotros por ejemplo queramos hacer un push para subir los archivos a GitHub, nos va a pedir un usuario y contraseña siempre, pero por lo menos no necesita de complejizar el proceso de crear una llave SSH y con cosas de seguridad.

[05:18] Entonces vamos a seguir lo que dice acá. Vamos a copiar este comando: "Ctrl + C", vamos a nuestra terminal, siempre estando en nuestra carpeta personal, Bruno en este caso que es donde tenemos el proyecto, vamos a copiar. Un segundo que copio lo que había seleccionado anteriormente. Copiar, clic derecho, pegar y vamos a analizar un poco el comando.

[05:50] Tenemos git remote que esa ya lo conocíamos. Add, git remote add y acá aparece origin. ¿Por qué origin? Origin es el repositorio remoto principal por defecto, por convención se eligió ese nombre, origin. Ustedes acá por ejemplo podrían cambiarlo por el nombre que ustedes quieran. Podrían poner si quisieran GitHub, pero yo voy a mantener esa convención para mantener un estándar.

[06:30] Y acá tenemos la dirección de nuestro repositorio, vamos a dar enter. Ahora el siguiente comando que nos dice que utilicemos es git branch -M main. Luego tenemos este git branch -M main. esto ha sido cambiado recientemente a la hora de crear este curso. Anteriormente no era necesario utilizar este comando para transformar nuestra rama master para el nombre main. Por una cuestiones políticas y sociales tal vez es que ha cambiado esto para main.

[07:14] Pero en nuestro caso, en este proyecto nosotros vamos a mantener por ahora el nombre master de nuestra branch principal. Entonces vamos a saltar ese comando y vamos a ir directamente para `git push -u origin` y acá vamos a cambiar por master. Entonces vamos a hacer eso de aquí y voy a explicarles el comando.

[07:38] `Git push` nosotros ya lo conocíamos de los videos anteriores. -u lo que hace es decirle a nuestro. Perdón, antes de explicar voy a cambiar acá a master que es el nombre de nuestra branch. Ahora sí, nuestro u significa que cada vez que nosotros hagamos un `git push` de nuestra branch master siempre va a ir a origin.

[08:04] Entonces nosotros por ejemplo podríamos colocar simplemente en el futuro `git push` nada más y estando nuestra rama master y enviaría directamente nuestro repositorio. En este caso yo prefiero sacar ese -u porque así me va a obligar a mí a estar poniendo siempre por ejemplo `git push origin master` y me permite tener un poco más de control de qué cosas estoy enviando para estar seguro de que lo que estoy haciendo, realmente está conforme a lo que quiero hacer.

[08:42] Entonces vamos a dejar simplemente `git push origin master`. Vamos a dar un enter, acá vamos a necesitar loguearnos. En el caso de que ustedes estén en otra plataforma que no sea Windows posiblemente no aparezca esa ventana y todo este navegador, simplemente ustedes van a necesitar colocar los datos dentro de la terminal propiamente.

[09:13] Una vez que nos logueamos en GitHub, nos va a pedir permisos para que unamos nuestro repositorio a GitHub. Vamos a dar Authorize. Ahora podemos cerrar nuestra ventana y se van a enviar los datos. Bien. Dice que se creó una nueva branch en GitHub y se envió a nuestra rama master.

[09:41] Vamos a volver a GitHub, vamos a dar un F5 para actualizar la página y acá tenemos los datos de nuestro repositorio. Vean que no se envió ni el

a.config ni la carpeta. ¿Por qué? Porque está dentro de nuestro git ignore. Una vez que nos hemos logueado y hemos hecho el refresh a la página, vemos que tenemos acá por ejemplo los archivos, tenemos acá en este icono los commits que hemos hecho, todos los commits que hemos hecho localmente.

[10:13] Por ejemplo si entramos en un commit en particular vemos qué archivo se modificó y qué líneas se han modificado, qué líneas se han quitado y cuáles se han agregado. Volvemos para atrás. GitHub nos provee una interfaz con muchas opciones y muchas cosas complejas. Podemos ver acá las ramas que tenemos.

[10:37] Nos está diciendo acá que tenemos una rama. Por ejemplo podríamos si quisiéramos agregar colaboradores, yendo a settings, manage access y acá podríamos invitar a colaboradores, otras personas. Si tuviéramos el email de Ana podríamos agregarla a nuestro repositorio.

[10:58] Como vemos, GitHub nos provee muchas, muchas opciones complejas que no vamos a ver ahora en este curso, quizás en un curso futuro veamos más cosas sobre GitHub, pero por ahora no nos vamos a adentrar demasiado en detalles en todas las opciones que nos ofrece. Entonces ahora ya podemos compartir código, sea a través de la red de nuestra casa o en mi computadora en este caso.

[11:23] O utilizando GitHub, ya podemos gestionar el código, las versiones de nuestro código de forma interesante. Pero existen formas un poco más rebuscadas, un poco más profesionales, de organizar nuestro sistema de control de versiones y vamos a hablar sobre branches, que ya hablamos un poco de master y todo esto en la próxima aula.

[11:43] Así que empiecen a hacer ejercicios, tiren todas las dudas que tengan y vamos para la próxima aula.